METHOD AND APPARATUS FOR PROCESSING FRAME CLASSIFICATION INFORMATION BETWEEN NETWORK PROCESSORS

Brian M. Bass, Jean L. Calvignac, Anthony M. Gallo, Marco Heddes, Michael S. Siegel and Fabrice Verplanken

BACKGROUND OF THE INVENTION

The present invention relates to network processors which transfer network data frames between nodes of a network. Specifically, a process and apparatus for supplying frame type information and information identifying the depth of processing of an ingress processor to an egress processor of a network node is described which avoids redundant frame processing.

Communication network systems transfer information between a source and destination in frames which are transferred between different nodes along the network. The frames contain data destined for the user as well as destination information and other information which is needed for later processing downstream. One of the more common devices for transferring frames of data between nodes is the multi-processor switch. The multiprocessor switch is organized as a series of blades for receiving traffic through multiple ports, and for delivering traffic through other ports on the same or different blades of the switch. The blades typically have an ingress processor associated with them which processes incoming frames by correlating the frame destination with an output port, determining from the frame various

parameters which are needed to process the frame, and transferring the processed frame to an egress processor on the same or a different blade. The egress processor may share the same hardware as the ingress processor and run a different set of pico code instructions when it is forwarding frames to an output port of the same blade.

Incoming frames are processed by the ingress processor by parsing the contents of the header, determining a destination for the frame, and then handing off the frame to the appropriate egress processor for processing and forwarding to an appropriate output port. The processing of frames by the ingress and egress processor necessarily requires redundant processing of the frame as it is handed off from the ingress processor to the egress processor. Parsing of header information from the received frame, identifying the frame type as well as other control parameters is part of a frame classification process executed in both the ingress and egress processor. The redundant processing of the frame produces delays in overall device throughput degrading the device performance. The present invention is directed to a system and method which will avoid redundant processing by the egress processor which has previously taken place in the ingress processor.

SUMMARY OF THE INVENTION

The present invention provides for the more efficient processing of frames which are being transferred within a network device such as

a network switch. The network device includes an ingress processor which receives the network frames on the input ports. The ingress processor parses the frame parameters and prepares an intra-switch frame for delivery to an egress processor of the switch. The intra-switch frame includes a frame header identifying the type of frame received from the input port, parameters which have already been determined by the ingress processor, a level of processing which has been conducted on the frame and a multicast/unicast control bit. The multicast/unicast control bit identifies to the egress processor whether or not one or more frames are to be prepared for multiple ports on the network switch. The intra-switch frame is passed to the egress processor which is associated with one or more output ports on the network switch, which transfers the frame to a port identified from information contained within the incoming frame. The egress processor completes processing of the frame from program instructions which may have a starting address identified by the header data, reflecting the level of processing carried out on the frame by the ingress processor, and from passed parameters which were previously determined by the ingress processor. In this way, the egress processor need not perform redundant processing of a frame which was previously carried out by the ingress processor.

In one embodiment of the invention, the frame header data is decoded in a hardware classifier logic circuit, while header data is stored in memory by the egress processor, to locate the starting

address for the egress processor.

In the other embodiments of the invention, the intra-switch frame header may include other control parameters determined by the ingress processor.

DESCRIPTION OF THE FIGURES

Figure 1 illustrates the architecture of a switch which receives incoming frames and transfers the incoming frame to an egress network processor;

Figure 2 illustrates the frame header data for the intra-switch frame prepared by the ingress processor;

Figure 3 is an illustration of the processor architecture for the egress processor of a network switch; and

Figure 4 illustrates the frame execution by the egress processor for intra-switch frames.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Network communications depend upon various components which can connect the segments of a network together such as network switches and bridges. Network switches may be organized in accordance with the architecture shown in Figure 1, wherein data frames transmitted on the network are received on a port of the switch. Each switch has multiple blades 3 which support multiple input ports 0-39, output ports 0-39, and a blade processor. As illustrated in Figure

1, input ports 0-39 associated with the first blade have an ingress processor 1 which receives the incoming frames to be routed by the switch. The received frames may be of different formats, depending on the various protocols served by the network switch. Processing of an incoming frame requires that the frame header information be parsed, and control parameters from the frame be derived to determine how the frame is to be processed.

Once the frame has been processed by the ingress processor 1, it can be handed off to the appropriate egress processor 2 which serves a port for transferring the frame to a destination address determined from the incoming frame. The egress processor 2 may be located on a different blade, or may be the ingress processor of the same blade, operating under a different set of pico code instructions.

The present invention takes advantage of the fact that a significant amount of the incoming frame processing which has already taken place in the ingress processor 1 need not be redundantly executed in the egress processor 2. The present invention simplifies processing for the egress processor 2 by passing an intra-switch frame containing data from the originating frame received by the switch, along with a header which contains control data prepared by the ingress processor 1. The control data contained in the intra-switch frame header may identify to the egress processor 2 the type of frame which was received from the network, parameters which were determined during processing by the ingress processor 1,

as well as a level of processing of the received frame by the ingress processor 1 in the form of data for locating a starting address for the egress processor pico code. From this data, it will be possible to identify a starting address for the egress processor 2 which avoids redundant processing of information which was determined by the ingress processor and is now contained in the intra-switch frame header.

The frame header for the intra-switch frame is more particularly shown in Figure 2. Referring now to Figure 2, the frame received from the network input port is shown as 11. A frame header 12 prepared by the ingress processor 1 includes a number of fields which will be used by the egress processor 2 to determine the starting point instruction for processing of the frame as well as additional parameters needed to complete the egress processing. A first field MC includes a one-bit index which identifies whether or not the frame being passed is a multicast or unicast frame. The field SP identifies the source port, the field SB identifies the source blade on which the frame entered the switch, and a parameter LID (lookup identifier) is in the following field.

The MC, variable software header format (VSHF), and frame header extension format (FHEF) are the main fields in the intra-switch frame header used to pass software execution control from the ingress processor 1 to egress processor 2.

When the variable header extension (VHE) field is used to specify control information, the VSHF field is programmed with data to

indicate the data type and format which would appear in the VHE field. When a fixed amount of control information is adequate, the FHEF field indicates the number of bytes in the following FHE field and its format which the egress processor 2 will use during frame processing. The capability of using the variable length header extension field VHE is available for yet unforeseen frame protocols which are to be received over the network to permit control information for such frames.

By decoding data in these fields, the egress processor 2 can begin processing the frame at a starting instruction which takes advantage of the earlier processing performed by the ingress processor. The starting instruction address location can be determined from data in the frame header by a hardware frame classifier as will be described with respect to the egress processor operation. In this way, the egress processor 2 can jump to a starting address which reflects the frame type which was identified by the ingress processor, as well as the level of execution completed by the ingress processor.

Sub B1

In accordance with the preferred embodiment, the frame header 12 of Figure 2 includes one bit in the MC field, two bits in the VFSH field, and two bits in the FHVF field which are exemplary only. The egress processor 2 has associated with it a hardware classifier which detects from the data in these three fields a five-bit code (one bit from the MC field, two bits each from the VFSH and FHVF fields).

The five-bit code identifies at least 32 starting addresses for 32 different sets of pico code to be executed by the egress processor.

The set of instructions identified by the five bit code may thereafter process data contained within the FHE, or VHE fields, as called for by the instructions being executed. This provides the ability for the intra-switch frame header 12 to pass to the egress processor not only starting instruction information to the egress processor to begin processing a set of instructions, but actual parameter values which may have been previously determined by the ingress processor.

Figure 3 illustrates the architectural organization of the egress processor 2 which receives the intra-switch frame. A frame input area FIA 6 defines a series of memory locations which receives the frame passed by the ingress processor. A copy of the first thirty-two bytes of the intra-switch frame shown in Figure 2 is stored in the frame input area (FIA). A copy of the port configuration entry for the interface on which the incoming frame to the switch was received is stored in a port configuration table 7. The port configuration table 7 can be used in a default circumstance, when configuration data in the egress processor turns the hardware classifier off, and execution of the frame begins as in the prior art, without any advance knowledge of the frame type, or the depth of processing previously executed by the ingress processor.

Processor 8 executes an instruction set for processing the data stored in the frame input area (FIA) 6. The Forwarding Enque Area

(FEA) 9 contains a list of ENQUE parameters, including well known frame alteration parameters (WFA) and flexible frame alteration parameters (FFA). Control structures are created for modifying the frame under control of the egress processor code. Processing of the frame is conventional in that search trees based on keys derived from the contents of the Frame Input Area (FIA) are created, and the results of that search produces other ENQUE parameters, as well as WFA and FFA which are stored in 9. Intermediate values obtained during these calculations are stored in the scratch memory 12 and general purpose registers 10. Hardware assist units 16 are also employed in the process for building the control structures in the Forwarding Enque Area.

In order to make use of the present invention, which seeks to avoid redundant processing by the egress processor 2, a hardware frame classifier 18 logic circuit is provided. The hardware frame classifier (HFC) 18 receives as information the control parameters from the header of Figure 2 on the intra-switch frame. As shown in Figure 2, the MC bit, the FHEF, VSHF bits from the header stored in frame input area FIA 6 are used to identify a starting instruction in processor 8 for commencing processing of the received frame. The decoding of the intra-frame header and selection of the instructions for commencing processing of the frame is shown in Figure 4.

Referring now to Figure 4, the frame header 12 for the intra-switch frame includes in the header a 2-bit extension format field

FHEF. The 2-bits in the extension format field FHEF can serve to identify a particular protocol layer in which the incoming frame was received by the switch. The values of the two bits and the identity of the protocol layer can be as set forth in Table 1.

Table 1:

00	=	FHE = 0 bytes, default - bridging
01	=	FHE = 4 bytes, IPV4
10	=	FHE = 4 bytes, SWJUMP
11	=	FHE = 4 bytes, encoded stake.

For the data entry 00, no data appears in FHE, such as when the egress processor is doing a bridging function.

For the data entry 01, the routing function is used by the egress processor and the IPV4 address for the next hop is inserted in the FHE field as four bytes of information. Thus, when decoding the header fields, the hardware frame classifier 18 utilizes the 2-bit information to identify the location of the 4 bytes and its meaning.

The data 10 indicates to the hardware frame classifier that four bytes of information within the FHE field is a starting address for the egress processor execution. When the hardware frame classifier 18 indexes the processor to the starting address, subsequent code execution by the processor will recognize that within the FHE are four bytes identifying a jump instruction to which the processor is to begin processing. In this way, starting instructions are not limited to 32, the

width of the 5-bit data, but may include others identified from the jump instruction.

The last entry in the table indicates the data entry 11 in the FHEF field identifies four bytes in the FHE field identifying the frame format. The FHE field contains the location of parameters forwarded by the ingress processor which may be used for processing the intra-switch frame having the identified format.

Two other bits which are provided to the hardware frame classifier are contained in the VSHF field. The VSHF field is related to the VHE field in that it specifies a number of bytes which appear in the VHE field in accordance with Table 2.

Table 2:

00	=	VHE = 0 bytes
01	=	VHE = 4 bytes
10	=	VHE = 6 bytes
11	=	VHE = first byte, VHE length.

These fields will permit further data generated from the ingress processor 1 to be inserted in the VHE field. Further, the data in the VHSF field can identify to the hardware frame classifier the parameter to which the data stored in the VHE field corresponds.

The two bits in the VSHF field can not only be decoded to indicate a particular parameter stored in the VHE field, but identifies the number of bytes in the VHE field containing the parameter. The

last entry in Table 2, 11, identifies to the hardware frame classifier 18 that the VHE length data is contained in the first byte of the VHE field.

The fifth bit is from the MC field which identifies the frame as either a multicast or unicast frame to the hardware frame classifier 18. The hardware frame classifier 18 includes a table which is indexed by the five bits input to the hardware frame classifier 18 from the foregoing MC, VHSF, FHEF fields. The five bits identify a starting address of the pico code instructions stored in the processor 8. Individual instructions following the starting address within the pico processor 8 may, in turn, look to the FHE fields, and VHE fields when the decoded output from the hardware frame classifier 18 points to the relevant set of instructions. For instance, when the FHE field is 10, a jump address is provided in the FHE field, which when the relevant instructions following the entry point of a set of instructions are executed, read the contents of the FHE field to obtain the jump instruction.

Likewise, other instructions which are pointed to as a result of decoding other combination of bits from the MC, VSHF, and FHEF fields by hardware classifier 18 will utilize parameters which have been stored in the FHE field or VHE field by the ingress processor, thus providing to the egress processor ready access to parameters previously computed by the ingress processor.

From the foregoing, the frame data is prepared in the forwarding enque area 9 for delivery to the output of the switch.

The egress processor can process multicast frames as well as unicast frames received from the ingress processor. The egress processor is associated with one or more output ports on the network switch and forwards the frame for transfer to a port identified from incoming destination information received from the incoming frame. When the egress processor wants to send a multicast frame out of several ports, it does so by queuing the multicast frame on each port the same way it would have been queued a unicast frame.

Thus, using the foregoing intra-switch frame, it is possible to pass off for execution control from the ingress processor to the egress processor so that redundant processing need not be carried out on the frame.

While the foregoing has been described with respect to a network switch, it should also be recognized that the principles are applicable within any network bridge, or in any network routing situation where frames must be processed a number of times by system components.

The foregoing description of the invention illustrates and describes the present invention. Additionally, the disclosure shows and describes only the preferred embodiments of the invention, but as aforementioned, it is to be understood that the invention is capable of use in various other combinations, modifications, and environments and is capable of changes or modifications within the scope of the inventive concept as expressed herein, commensurate with the above

teachings, and/or the skill or knowledge of the relevant art. The embodiments described hereinabove are further intended to explain best modes known of practicing the invention and to enable others skilled in the art to utilize the invention in such, or other, embodiments and with the various modifications required by the particular applications or uses of the invention. Accordingly, the description is not intended to limit the invention to the form disclosed herein. Also, it is intended that the appended claims be construed to include alternative embodiments.